

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Applicant(s): YAMADA, et al.
Serial No.: Not yet assigned
Filed: July 7, 2003
Title: DATA FORMAT CONVERSION METHOD AND EQUIPMENT,
AND CONTROLLER MANAGEMENT SYSTEM USING DATA
FORMAT CONVERSION EQUIPMENT
Group: Not yet assigned

LETTER CLAIMING RIGHT OF PRIORITY

Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

July 7, 2003

Sir:

Under the provisions of 35 USC 119 and 37 CFR 1.55, the applicant(s) hereby claim(s) the right of priority based on Japanese Patent Application No.(s) 2002-198773, filed July 8, 2002.

A certified copy of said Japanese Application is attached.

Respectfully submitted,

ANTONELLI, TERRY, STOUT & KRAUS, LLP



Carl I. Brundidge
Registration No. 29,621

CIB/alb
Attachment
(703) 312-6600



日 本 国 特 許 庁
JAPAN PATENT OFFICE

別紙添付の書類に記載されている事項は下記の出願書類に記載されている事項と同一であることを証明する。

This is to certify that the annexed is a true copy of the following application as filed with this Office

出 願 年 月 日

Date of Application:

2002年 7月 8日

出 願 番 号

Application Number:

特願2002-198773

[ST.10/C]:

[JP2002-198773]

出 願 人

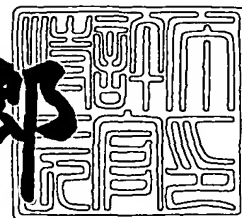
Applicant(s):

株式会社日立製作所

2003年 4月22日

特 許 庁 長 官
Commissioner,
Japan Patent Office

太田 信一郎



出証番号 出証特2003-3029531

【書類名】 特許願

【整理番号】 PE28646

【提出日】 平成14年 7月 8日

【あて先】 特許庁長官 殿

【国際特許分類】 G05B 19/048
G05B 19/05
G06F 9/06

【発明者】

【住所又は居所】 茨城県日立市大みか町七丁目1番1号
株式会社日立製作所 日立研究所内

【氏名】 山田 勉

【発明者】

【住所又は居所】 茨城県日立市大みか町七丁目1番1号
株式会社日立製作所 日立研究所内

【氏名】 齊藤 雅彦

【発明者】

【住所又は居所】 茨城県日立市大みか町七丁目1番1号
株式会社日立製作所 日立研究所内

【氏名】 鈴木 昭二

【発明者】

【住所又は居所】 茨城県日立市大みか町七丁目1番1号
株式会社日立製作所 日立研究所内

【氏名】 遠藤 浩通

【発明者】

【住所又は居所】 茨城県日立市大みか町七丁目1番1号
株式会社日立製作所 日立研究所内

【氏名】 松本 典剛

【発明者】

【住所又は居所】 茨城県日立市大みか町五丁目2番1号

株式会社日立製作所 情報制御システム事業部内

【氏名】 笠嶋 広和

【発明者】

【住所又は居所】 茨城県日立市大みか町五丁目2番1号

株式会社日立製作所 情報制御システム事業部内

【氏名】 山内 学

【特許出願人】

【識別番号】 000005108

【氏名又は名称】 株式会社日立製作所

【代理人】

【識別番号】 100098017

【弁理士】

【氏名又は名称】 吉岡 宏嗣

【手数料の表示】

【予納台帳番号】 055181

【納付金額】 21,000円

【提出物件の目録】

【物件名】 明細書 1

【物件名】 図面 1

【物件名】 要約書 1

【ブルーフの要否】 要

【書類名】 明細書

【発明の名称】 プログラム等のデータ形式変換方法及び装置、並びにそのデータ形式変換装置を用いたコントローラ管理システム

【特許請求の範囲】

【請求項 1】 第 1 データ形式のプログラム等を含む入力情報に含まれるデータ形式の変換指令で指定される変換規則に基づいて、第 1 データ形式のプログラム等を第 2 データ形式のプログラム等に変換するデータ形式変換方法。

【請求項 2】 前記変換指令は、少なくとも 1 つの第 1 データ形式に対応させて複数の異なる第 2 データ形式に変換する複数の変換規則の 1 つを指定するものであることを特徴とする請求項 1 に記載のデータ形式変換方法。

【請求項 3】 前記変換規則は、通信ネットワークを介して取得することを特徴とする請求項 1 に記載のデータ形式変換方法。

【請求項 4】 通信ネットワークから第 1 データ形式のプログラム等を含む入力情報を取り込む受信手段と、取り込まれた入力情報に含まれるデータ形式の変換指令を抽出する解釈手段と、該抽出された変換指令に対応するデータ形式の変換規則を取得する変換データ取得手段と、取得された変換規則に基づいて第 1 データ形式のプログラム等を第 2 データ形式のプログラム等に変換する変換実行手段と、該変換されたプログラム等を通信ネットワークに出力する送信手段とを構成する計算機を備えてなるデータ形式変換装置。

【請求項 5】 第 1 データ形式のプログラム等を第 2 データ形式のプログラム等に変換する変換規則が格納された記憶手段を備え、前記記憶手段は、少なくとも 1 つの第 1 データ形式に対応させて複数の異なる第 2 データ形式に変換する複数の変換規則を記憶してなり、前記変換データ取得手段は、前記記憶手段から前記変換指令に対応した変換規則を取得することを特徴とする請求項 4 に記載のデータ形式変換装置。

【請求項 6】 前記変換データ取得手段は、前記受信手段を介して外部に設けられた記憶手段に格納された前記変換規則を前記通信ネットワークから取得することを特徴とする請求項 5 に記載のデータ形式変換装置。

【請求項 7】 前記外部に設けられた記憶手段は、前記通信ネットワークに

通信可能に接続された記憶手段又は前記通信ネットワークに通信可能に接続された組み込み計算機を有してなるコントローラに備えられた記憶手段であることを特徴とする請求項 6 に記載のデータ形式変換装置。

【請求項 8】 前記変換規則は、第 1 データ形式を第 2 データ形式に変換する変換プログラムであり、前記変換実行手段は前記変換プログラムに従って第 1 データ形式のプログラム等を第 2 データ形式に変換することを特徴とする請求項 4 乃至請求項 7 のいずれかに記載のデータ形式変換装置。

【請求項 9】 通信ネットワークから第 1 データ形式のプログラム等を含む入力情報を取り込む通信手段と、取り込まれた入力情報に含まれるデータ形式の変換指令を抽出する解釈手段と、該抽出された変換指令に対応するデータ形式に変換する変換プログラムを前記通信手段を介して外部から取得する変換データ取得手段と、取得された変換プログラムに従って第 1 データ形式のプログラム等を第 2 データ形式に変換して前記通信手段を介して前記通信ネットワークに出力する変換実行手段とを構成する計算機を備えてなり、前記変換プログラムは第 1 と第 2 のデータ形式とは異なる第 3 データ形式に変換した後、第 2 のデータ形式に変換するものであるデータ形式変換装置。

【請求項 10】 前記変換データ取得手段は、前記受信手段を介して外部に設けられた記憶手段にアクセスして前記変換変換プログラムを前記通信ネットワークから取得することを特徴とする請求項 9 に記載のデータ形式変換装置。

【請求項 11】 前記変換実行手段は、仮想機械と第 3 データ形式に変換する中間変換手段とから構成されてなることを特徴とする請求項 9 又は 10 に記載のデータ形式変換装置。

【請求項 12】 前記仮想機械は、Java の仮想機械であることを特徴とする請求項 11 に記載のデータ形式変換装置。

【請求項 13】 第 1 データ形式によりプログラム等を作成し、該作成したプログラム等を第 2 データ形式に変換する指令を付加したメッセージを作成する計算機を備えてなるプログラミング装置。

【請求項 14】 第 1 データ形式のプログラム等を作成するプログラム作成手段と、該作成されたプログラム等を第 2 データ形式に変換する指令を付加した

メッセージを通信ネットワークに出力する通信手段とを構成する計算機を備えてなるプログラミング装置。

【請求項15】 第1データ形式のプログラム等を第2データ形式に変換する変換指令を入力する入力手段を備えてなることを特徴とする請求項14に記載のプログラミング装置。

【請求項16】 前記計算機は、携帯通信端末に組み込まれた計算機であることを特徴とする請求項13乃至15のいずれかに記載のプログラミング装置。

【請求項17】 通信ネットワークを介して入力される第1データ形式のプログラム等を格納する記憶手段と、通信ネットワークを介して入力される要求に応答して前記記憶手段に格納された第1データ形式のプログラム等を前記通信ネットワークに出力する機能を構成する計算機を備え、予め定められた第2データ形式のプログラム等に従って制御対象を制御するコントローラ。

【請求項18】 記憶手段に格納されている第2データ形式の制御プログラム等に従って制御対象を制御する計算機を備えたコントローラと、該コントローラに通信ネットワークを介して通信可能に設けられたコントローラ管理装置とを備え、該コントローラ管理装置は、前記コントローラの制御プログラム等を第1データ形式により作成するプログラミング装置と、該プログラミング装置により作成された第1データ形式のプログラム等を入力し、該プログラム等を第2データ形式のプログラム等に変換して前記コントローラに出力するデータ形式変換装置とを有してなるコントローラ管理システム。

【請求項19】 前記プログラミング装置と前記データ形式変換装置は、通信ネットワークを介して接続されていることを特徴とする請求項18に記載のコントローラ管理システム。

【請求項20】 前記プログラミング装置は、第1データ形式のプログラム等を第2データ形式に変換する変換指令を入力する入力手段を備え、該変換指令を第1データ形式のプログラム等に付加したメッセージを前記データ形式変換装置に出力することを特徴とする請求項18又は19に記載のコントローラ管理システム。

【請求項21】 第2データ形式の制御プログラム等に従って制御対象を制

御する計算機を備えたコントローラを管理する管理方法において、前記コントローラの制御プログラム等を第1データ形式により作成し、該作成された第1データ形式のプログラム等を第2データ形式のプログラム等に変換するデータ形式変換装置を介して前記コントローラに出力するコントローラ管理方法。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】

本発明は、プログラム等のデータ形式を異なるデータ形式に変換するデータ形式変換方法及び装置に係り、特に組み込み電子計算機（以下、単に計算機と略す。）を有するコントローラを管理するコントローラ管理システム、特にコントローラを遠隔から管理するコントローラ管理システムに好適なデータ形式変換方法及び装置に関する。

【0002】

本明細書において、プログラム等とはプログラム及びプログラムに準ずる電子情報をいう。また、プログラムとは計算機に対する指令であって、一の結果を得ることができるように組み合わせられたものをいい、プログラムに準ずる電子情報とは計算機による処理の用に供する情報をいう。また、データ形式とは、プログラム等の言語及び記述、データ構造等の形式をいう。また、コントローラの管理とは、コントローラに組み込まれた計算機のプログラム等の作成、編集、書換えなどの管理の他、通信ネットを介して遠隔からコントローラを監視したり、制御する管理を含むものとする。

【0003】

【従来の技術】

プログラマブルコントローラやプログラマブルロジックコントローラ（PLC）等のような計算機が組み込まれたコントローラ（以下、単にコントローラと称する。）が広く使用されている。このようなコントローラの計算機にプログラムをインストールしたり書換える場合、専用のプログラミング装置を組み込み計算機にケーブル等で接続して、専用プログラミング装置で作成したしたプログラムをその計算機に転送するのが一般的である。

【0004】

しかし、計算機の技術革新や進歩に起因して専用プログラミング装置の機能の陳腐化が激しく、各種のコントローラに対応する専用プログラミング装置を確保しておくことが困難であるという問題がある。すなわち、社会基幹施設を制御するコントローラの中には、15年以上の長期にわたって運用されるものがあるから、それらのコントローラの管理も長期にわたって続けなければならない。このような場合、従来では、15年間管理できるように故障を予測した上で専用プログラミング装置を予め確保しておく必要があり、費用の効率性や性能改善の点にも問題がある。

【0005】

このような問題を解決するため、特開平11-24716号公報（以下、先行技術1という。）に、汎用性やコストパフォーマンスに優れたパーソナルコンピュータ（PC）、及びインターネットに代表される通信アプリケーション技術を利用した管理システムが提案されている。これによれば、ユーザのコントローラにHTTP（Hyper Text Transfer Protocol）サーバを搭載し、プログラマーはPCのWebブラウザを介してコントローラからHTML（Hyper Text Markup Language）データを取得し、取得したHTMLデータに従ってコントローラのプログラムを制御するようにしている。

【0006】

また、コントローラの管理を効率化するために、プログラムの記述言語等を標準化することが要求されている。この要求を満たすために、米国特許公開公報（US2002/0004804A1）（以下、先行技術2という。）においては、プログラムをマークアップ言語で記述することが提案されている。マークアップ言語の例としてXML（eXtensible Markup Language）を取り上げ、プログラミング装置はエディタで作成されたプログラムをXML形式のプログラムに変換してコントローラに転送するようにしている。そして、コントローラは、XML形式のプログラムを自身の記憶装置に格納し、そのプログラムをコントローラ内部の制御実行手段により解釈して制御対象を制御するようにしている。なお、XMLは、標準化団体であるW3C（World Wide Web Consortium）にて策定されているデータ規格

である。今後ネットワーク上で流通するデータの標準として、長期にわたり利用できるものとして有望視されている。

【0007】

一方、特開2001-350507号公報（以下、先行技術3という。）に、人とインターフェースをとるためのブラウザを内蔵する通信端末に関し、性能が限定される通信端末においてマークアップ言語で記述されたデータを閲覧する技術が開示されている。すなわち、WML（Wireless Markup Language）で記述されたデータを通信端末がHTTPサーバから取得する経路に、ネットワークインターフェースを介在させている。このネットワークインターフェースは、HTTPサーバから取得したWMLソースを通信端末に送信する前に、コンパイル済みWMLのコンテンツに変換するWAPゲートウェイ手段を有する。そして、通信端末のユーザがHTTPサーバにWML情報を要求すると、ネットワークインターフェースでWMLソースをコンパイルして、その結果を通信端末に送信する。これにより、処理能力が限定された通信端末においても、マークアップ言語で記述されたデータを閲覧することが可能となる。

【0008】

一方、半導体プロセス技術と通信ネットワーク技術の進歩により、従来では対応できなかった分野あるいは製品に、通信ネットワークに接続された計算機を組み込むことが行なわれてきている。このような組み込み計算機は、稼動時のコストを最小とすることが重要であることから、組み込み時点において機能を限定し、拡張性が少ないことも一般的である。つまり、計算機の機能を特定の制御に特化して組み込まれている。このような組み込み計算機と、通信ネットワーク及び通信端末を用いることで、時間と場所を選ばず監視等を含む制御を行うサービスが立ち上がりつつある。

【0009】

【発明が解決しようとする課題】

しかしながら、通信ネットワークから組み込み計算機をプログラミングしたり、監視制御するには解決しなくてはならない課題が存在する。すなわち、先行技術1に提案されたコントローラ管理システムは、コントローラに格納されている

HTMLデータの内容に制御が限定されるという問題がある。

【0010】

この問題を解決する方法として、例えば、HTTPサーバの機能を拡張する技術であるCGI (Common Gateway Interface) を利用することが考えられる。

【0011】

しかし、この場合は、HTTPサーバにCGI機能を持たせるとともに、十分な処理能力を持たせる必要があるが、組み込み計算機にCGI機能や処理能力を持たせるのは現実的でない。また、必要な機能を追加する度に、CGIプログラムをHTTPサーバに登録する必要があるから柔軟性に欠ける。また、サーバ機能を拡張する技術として、CGIの他にSSI (Server Side Include)、JSP (Java (登録商標) Server Page) 等があるが、同様の問題がある。

【0012】

他方、先行技術2に提案されたコントローラ管理システムは、プログラムとしてマークアップ言語を用いることから、記述されているデータ構造がデータ定義と一致するかを検査する妥当性検査が必要となる。特に、XMLでは妥当性検査が必要である。また、XMLはテキストで記述されているので、組み込み計算機は直接XMLを利用できない。そこで、組み込み計算機が利用可能な情報を、テキスト記述から抽出する処理が必要となる。

【0013】

しかし、これらの処理は一時的な処理であり、稼動時のコスト最小が優先される組み込み計算機においては、その処理のために計算処理能力を高めるのは問題がある。また、プログラミング装置としては、コントローラ毎に対応するXML変換手段を用意する必要があり、メーカーや種類が異なるコントローラに適用するのは困難である。

【0014】

さらに、先行技術3に開示されるゲートウェイ装置は、ソースからコンパイルされる言語への対応付けが一意に決定されてしまう。

【0015】

上述したように、多種多様な組み込み計算機を有するコントローラは、対応で

きる言語が個別に定まっていることから、多種多様な組み込み計算機を有するコントローラに対して、通信ネット等を介してプログラミングを行なうことができる柔軟性のある汎用性の高いプログラミング装置の開発が要望されている。

【 0 0 1 6 】

そこで、本発明は、プログラム等のデータ形式を異なるデータ形式に変換するデータ形式変換装置を提供することを第 1 の課題とする。

【 0 0 1 7 】

また、本発明は、コントローラを管理するシステムの汎用性ないし柔軟性を高めることを第 2 の課題とする。

【 0 0 1 8 】

また、本発明は、コントローラを遠隔から管理制御するシステムを実現することを第 3 の課題とする。

【 0 0 1 9 】

【課題を解決するための手段】

本発明は、次に述べる手段により、上記課題を解決するものである。

【 0 0 2 0 】

本発明に係るデータ形式変換装置は、第 1 データ形式のプログラム等を含む入力情報を取り込み、該入力情報に含まれるデータ形式の変換指令に基づいて、第 1 データ形式のプログラム等を第 2 データ形式のプログラム等に変換して出力することを特徴とする。つまり、第 1 データ形式のプログラム等を作成するプログラマーは、その作成されたプログラム等を第 2 データ形式に変換する指令を付加したメッセージをデータ形式変換装置に入力することが特徴である。

【 0 0 2 1 】

このような本発明のデータ形式変換装置によれば、例えば、コントローラの制御プログラムを第 2 データ形式とすると、プログラマー作成者は、可読性を有し、かつ計算機と親和性が高く、プログラミング作業が容易なマークアップ言語等の第 1 データ形式で制御プログラムを作成すれば、自動的に第 2 データ形式の制御プログラムに変換することができる。その結果、プログラマー作成者は第 2 データ形式に習熟することなく、しかもコントローラに適合した専用プログラミング装

置が陳腐化していたとしても、既設コントローラの制御プログラムの作成、変更、編集、書き換え、等の管理を容易に行なうことができる。しかも、コントローラのデータ形式と異なる任意のデータ形式でプログラム等を作成することができるから、汎用性がありオープン性の高い計算機（例えば、パーソナルコンピュータ）を用いることができるから、コントローラを管理するシステムの汎用性ないし柔軟性を高めることができる。

【 0 0 2 2 】

また、本発明のデータ形式変換装置は、第 1 データ形式のプログラム等を第 2 データ形式のプログラム等に変換する変換規則が格納された記憶手段を備え、前記記憶手段は、少なくとも 1 つの第 1 データ形式に対応させて複数の異なる第 2 データ形式に変換する複数の変換規則を記憶してなるものとすることができる。

【 0 0 2 3 】

この場合において、前記記憶手段を前記データ形式変換装置の外部に設け、通信手段を介して前記記憶手段から前記変換規則を取得するようにすれば、遠隔から同種又は複数種類の複数のコントローラを管理することができる。

【 0 0 2 4 】

本発明のデータ形式変換装置は、具体的には、通信ネットワークから第 1 データ形式のプログラム等を含む入力情報を取り込む受信手段と、取り込まれた入力情報に含まれるデータ形式の変換指令を抽出する解釈手段と、該抽出された変換指令に対応するデータ形式の変換規則（例えば、変換プログラム）を取得する変換データ取得手段と、取得された変換規則に基づいて第 1 データ形式のプログラム等を第 2 データ形式のプログラム等に変換する変換実行手段と、該変換されたプログラム等を通信ネットワークに出力する送信手段とを構成する計算機により実現できる。そして、第 1 データ形式のプログラム等を第 2 データ形式のプログラム等に変換する変換規則が格納された記憶手段を備え、前記記憶手段は、少なくとも 1 つの第 1 データ形式に対応させて複数の異なる第 2 データ形式に変換する複数の変換規則を記憶してなり、前記変換データ取得手段は、前記記憶手段から前記変換指令に対応した変換規則を取得するようにすることができる。

【 0 0 2 5 】

この場合において、前記変換データ取得手段は、前記受信手段を介して外部に設けられた記憶手段に格納された前記変換規則を前記通信ネットワークから取得することができる。また、外部に設けられた記憶手段としては、前記通信ネットワークに通信可能に接続された記憶手段又は前記通信ネットワークに通信可能に接続された組み込み計算機を有してなるコントローラに備えられた記憶手段のいずれであってもよい。

【 0 0 2 6 】

上記の場合において、前記変換プログラムは第2のデータ形式とは異なる第3データ形式に変換した後、第2データ形式に変換するものとすることができる。この場合の変換実行手段は、仮想機械と第3データ形式に変換する中間変換手段とから構成することができる。また、仮想機械は、Javaの仮想機械とすることができる。

本発明に係るプログラミング装置は、第1データ形式のプログラム等を第2データ形式に変換する変換指令を入力する入力手段を備え、データ形式変換装置に送出するメッセージに付加することができる。この場合のプログラミング装置は、携帯通信端末に組み込まれた計算機により実現できる。

【 0 0 2 7 】

また、予め定められた第2データ形式のプログラム等に従って制御対象を制御するコントローラを、通信ネットワークを介して入力される第1データ形式のプログラム等を格納する記憶手段と、通信ネットワークを介して入力される要求に応答して前記記憶手段に格納された第1データ形式のプログラム等を前記通信ネットワークに出力する機能を構成する計算機を備えて構成することができる。これによれば、プログラミング装置により作成された第1データ形式のプログラム等がコントローラに保存されるので、プログラミング装置に第1データ形式のプログラム等を保存する必要がないばかりでなく、異なる任意のプログラミング装置により当該コントローラのプログラム等の管理をすることができる。

【 0 0 2 8 】

本発明のコントローラ管理方法及びシステムは、上述したプログラミング装置及びデータ形式変換装置からなるコントローラ管理装置を、対象のコントローラ

に直接又は通信ネットワークを介して接続することにより実現できる。

【 0 0 2 9 】

特に、コントローラが家庭電化製品を制御対象とする場合は、通信ネットワークに家庭電化製品を制御対象とするコントローラを通信可能に接続し、携帯電話などの携帯通信端末をプログラミング装置とし、そのプログラミング装置からデータ形式変換装置を呼出してコントローラに制御プログラムを送ることにより、携帯電話から自己の家庭の電化製品の制御プログラムを変更等することができる。

【 0 0 3 0 】

【発明の実施の形態】

以下、本発明を実施の形態に基づいて説明する。

（第1の実施の形態）

図1に、本発明の第1実施形態に係るコントローラ管理システムの全体構成を示す。図示のように、本発明によるコントローラ管理システムは、プログラミング装置100とデータ形式変換装置200とからなる管理装置と、管理対象のコントローラ300と、コントローラ300により制御される制御対象400を含んで構成される。

【 0 0 3 1 】

プログラミング装置100は、コントローラ300が実行するプログラム等を作成するものであり、そのプログラム等を作成するエディタ110を有して構成される。ここで、プログラム等とは、先に定義したとおり、プログラム及びプログラムに準ずる電子情報をいい、計算機による処理の用に供する全ての情報を含むものとする。プログラミング装置100は、エディタ110により作成されたプログラム等からなるコントローラの管理制御情報（以下、メッセージと称する）を作成してデータ形式変換装置200に出力するようになっている。なお、コントローラの管理制御情報であるメッセージには、プログラム等の他に、管理対象のコントローラ300の種類を特定するコントローラタイプ、コントローラ300のアドレス等の識別データ、その他の情報が含まれている。

【 0 0 3 2 】

ここで、プログラミング装置100は、パーソナルコンピュータ（PC）等の汎用性及びオープン性の高い計算機を用いて構成することが、将来の部品調達の観点から望ましい。また、エディタ110は、PCで動作するプログラムにより実現するのが好適である。しかし、プログラミング装置100は、これらの組み合わせに限定されるものではなく、例えばエディタプログラムとしてPCのWebブラウザで動作するWebアプリケーションを適用することができる。これによれば、Webブラウザの使い勝手と、実行プラットフォーム独立性とを享受することができる。また、入出力点数が少ないコントローラ300の場合（例えば、入出力点数が数十個以内）、プログラミング装置100は携帯電話のようなボタンと簡易な表示器の組み合わせで構成することも可能である。

【0033】

データ形式変換装置200は、解釈制御部210と、変換データ取得部220と、変換実行部230と、記憶装置240を含んで構成されている。解釈制御部210は、プログラミング装置100から受信するメッセージのデータ構造が、妥当なものであるかを検査する機能と、メッセージから適宜目的情報を抽出する機能を有してなる。例えば、メッセージのデータ形式がXML形式の場合には、解釈制御部210にXMLパーサを備えてデータ構造の妥当性を判断する。変換データ取得部220は、解釈制御部210により抽出されたコントローラタイプ情報に従って、指定されたコントローラタイプに対応する変換プログラムを記憶装置240から読み出して変換実行部230に転送する。変換実行部230は、変換データ取得部220により読み出された変換プログラムに基づいて機能が設定され、あるいは起動され、解釈制御部210で受信したプログラム等をコントローラ300に適合するプログラム等に変換する。つまり、メッセージのプログラム等のデータ形式を、コントローラ300に適合したデータ形式のプログラム等に変換する。記憶装置240には、本実施形態の管理システムが対象とする各種のコントローラのデータ形式に対応させて変換プログラムが格納されている。

【0034】

コントローラ300は、データ形式変換装置200から出力されるメッセージを受信して記憶装置301に格納する。コントローラ300は、予め設定されて

いる動作又は受信したメッセージに従って制御対象400を制御する。制御対象400は、工場や生産ライン、基幹システム、家庭電化製品等が該当する。制御対象400の具体例としては、入出力接点、アナログ入出力、フィールドネットワーク、サーボアンプ等の機械装置、家庭におけるエアコンや洗濯機、監視カメラ等の一般家電等、制御要素が該当する。特に、一般家電を制御するインターフェースとしては、赤外線、無線、電灯線、専用制御線などを活用することができる。

【0035】

このように、従来のプログラミング装置は、コントローラのデータ形式に適合したプログラム等を作成してコントローラに直接転送していたのに対し、図1の実施形態のプログラミング装置100は、任意のデータ形式でプログラム等を作成し、作成されたプログラム等はデータ形式変換装置200によりコントローラ300に適合したデータ形式に変換されて、コントローラ300に転送される。つまり、本実施形態では、プログラミング装置100において、コントローラ300を制御するためのプログラム等からなるメッセージが作成される。メッセージを構成するデータ形式又は構造としては、バイナリデータやテキストデータがあるが、テキストデータとしてXML形式のデータ構造を採用することが好ましい。つまり、XML形式は冗長であるが、機械語に比べて人が読んで理解できる可読性に優れているとともに、PC等の計算機にて扱いやすい。以下、XML形式のメッセージを例として本実施形態の説明を行なうが、本発明はXML形式に限定されないのは言うまでもない。

【0036】

図2に、プログラミング装置100から出力されるXML形式で記述されたメッセージの一例を示す。図において、行頭の数字とコロンは説明のために付与したものであり、XML形式のメッセージを構成するものではない。なお、XMLに関する文法的な規定は、W3Cが公開するWebサイト（URL=<http://www.w3.org/>）、『よくわかるXMLの基礎』（日経BP社刊、原著：Simon St. Laurent、訳者：藤本淑子）等の文献に詳しいので詳細な説明は省略する。

【0037】

図 2 において、1 行目は XML 宣言であり、本メッセージが XML 形式で記述されていることを表している。2 行目は文書型宣言であり、本メッセージのデータ構造が "someURI/plc.dtd" で表される文書型定義 (Document Type Definition; DTD と略す) ファイルに定義されていることを示す。解釈制御部 210 は、ここで指定される DTD を取り込み、受信したメッセージのデータ構造が DTD に適合するか否か検査する。3 行目はメッセージの本体部分を示す開始タグであり、12 行目の終了タグと対をなす。メッセージ本体は、さらに 2 つに分かれており、4 行目から始まる記述と、9 行目から始まる記述からなる。4 行目はプログラムを補足するデータ部の開始タグであり、8 行目の終了タグと対をなす。5 行目は、本メッセージが変換される対象のコントローラタイプが記述される。6 行目は、本メッセージの送信先であるコントローラの宛先のアドレスが記述される。7 行目は、本メッセージが発信された送信元の識別コードが記述される。本例では、6、7 行目の宛先と送信元として IP アドレスで表した例を示した。しかし、これはメッセージが伝達されるネットワークメディアに依存するものであり、装置を特定できる情報であればこれに限られない。望ましくは、タグの属性としてデータのタイプを指定する。例えば、IPv4 アドレスであれば <DEST type="IPv4"> と記述する。9 行目はプログラム本体の開始タグであり、11 行目と対を成す。10 行目では、コントローラ 300 を制御するためのプログラムが記述される。本例では詳細な説明を省略するが、例えば SOAP (Simple Object Access Protocol) で規定されるように、オブジェクトに対する操作を列挙することが可能である。

【0038】

図 2 に示すデータ形式を採用することにより、処理されるべき内容を指定できる共に、コントローラ 300 を制御するプログラム等を記述できる。また、人による可読性と計算機によるデータ可用性の両立を図ることが可能となる。なお、図 2 に示されたデータ形式において、必要な情報が存在する限り、データの追加や削除、変更は可能である。例えば、メッセージのデータ形式が整形形式 (well formed) のみの場合には、2 行目の文書型宣言は削除可能である。

【0039】

なお、メッセージが整形形式である場合には、解釈制御部 2 1 0 は非妥当性検証パーサを用いて、メッセージが整形形式であるか否かを検証することが望ましい。また、メッセージが図 2 の 2 行目のスキーマ定義を含む場合には、解釈制御部 2 1 0 は妥当性検証パーサを用いて、メッセージがスキーマ定義と合致するか否かを検証する必要がある。

【 0 0 4 0 】

ここで、図 3 に図 2 のメッセージの変形例を示す。同図に示したメッセージは、プログラミング装置 1 0 0 から出力される XML 形式で記述されたメッセージの一部であり、図 2 の 5 行目の部分に相当する。この例では、データ形式変換装置 2 0 0 におけるデータ形式の変換を確実にを行うために、変換元のデータ形式あるいは変換後のデータ形式を、メッセージにより指定するようにしている。すなわち、変換元のデータ形式を"SRC"属性で指定し、変換後のデータ形式を"DEST"属性で指定している。この例では、変換元形式が"XML-C"という型、変換後形式が"MAC-4"という型で、コントローラタイプがEQ-XYというコントローラに対する指定を意味している。なお、これらの記号は説明用のために任意に決めた記号である。

【 0 0 4 1 】

図 3 に示したように、メッセージの変換元形式あるいは変換後形式として複数のデータ形式を選択できる場合は、そのデータ形式を指定するために属性指定することが好適である。なお、変換元形式あるいは変換後形式が複数存在するにもかかわらず、属性指定が存在しないメッセージが入力された場合は、予め定めたデータ形式を適用して変換するか、あるいはエラーとして変換しないことが望ましい。ここで、設定されたデータ形式を適用したり、エラーとして処理するには、例えばDTDや他のスキーマ指定により設定することが可能である。

【 0 0 4 2 】

ここで、図 4 に、変換元のメッセージと変換後のメッセージのデータ形式を対比して示す。同図は、XML 形式のメッセージの変換を説明する図であり、タイプ名 1 2 0、管理対象の対象コントローラ 1 2 1、変換元形式 1 2 2、変換後形式 1 2 3 を示している。同図に示すように、タイプCTL-Aの場合は、「A型

コントローラ」に適するように、変換元形式「MAC-1」から変換後形式「MAC-2」に変換する。また、タイプEQ-XYの場合は、「XY仕様に準拠する装置」に適するように、変換元形式「MAC-3」、「XML-C」、「LANG-D」のいずれかから、変換後形式「MAC-4」、「XML-E」のいずれかの形式に変換する。

【0043】

ここで、変換元あるいは変換後のデータ形式として、図中「MAC-」として表記したコントローラ固有の機械語、図中「XML-」として表記した特定分野又は団体で合意されているXML形式、あるいは図中「LANG-」として表記した特定の形式を挙げることができる。また、これらの他に、IEC (International Electro-technical Commission) 61131-3規格で定められるプログラムの記述言語IL (Instruction List) 又はST (Structured Text) などが挙げられる。

【0044】

図5に、図1の実施形態における処理流れのシーケンス図を示す。同シーケンス図では、図の上から下に処理時間経過を示している。初めに、プログラミング装置100からデータ形式変換装置200に対してメッセージが送信される（事象500）。データ形式変換装置200では、解釈制御部210がメッセージを受信する。受信したメッセージが妥当なものであるか検査するために、解釈制御部210でメッセージ解釈処理を実行する（事象501）。ここではメッセージで指定されるDTDに従い妥当性検証が行われ、さらにXMLデータから必要な情報を抽出する。次に、変換データ取得部220に対しメッセージの変換タイプを通知する（事象502）。変換データ取得部220は通知された変換タイプによって直接的あるいは間接的にメッセージの変換方法を判定する（事象503）。例えば、変換タイプと同一のファイル名で格納される変換プログラムを記憶装置240より取得する。あるいは、変換データ取得部220内部で持つデータベースから変換タイプと変換プログラムの対応関係を取得した後、記憶装置240より変換プログラムを取得する。その後、変換データ取得部220は変換実行部230の設定を行う（事象504）。例えば、変換タイプに対応した変換プログ

ラムを起動させて変換実行部 2 3 0 を構成させる。ここで、変換実行部 2 3 0 は、変換データ取得部 2 2 0 により起動させてもよく、また予め起動させてもよい。変換実行部 2 3 0 の準備が整い次第、変換実行部 2 3 0 から変換データ取得部 2 2 0 へ設定完了を通知する（事象 5 0 5）。通知を受けて、変換データ取得部 2 2 0 は変換タイプの指定完了を解釈制御部 2 1 0 に通知する（事象 5 0 6）。解釈制御部 2 1 0 は、受信したメッセージ又はそのプログラム部分を変換実行部 2 3 0 に転送する（事象 5 0 7）。変換実行部 2 3 0 は、メッセージのプログラム部分を、変換データ取得部 2 2 0 により取得された変換プログラムに従って変換し、その変換されたメッセージをコントローラ 3 0 0 に送信する（事象 5 0 8）。コントローラ 3 0 0 は変換されたメッセージを受信して記憶装置 3 0 1 に格納する。そして、コントローラ 3 0 0 は記憶装置 3 0 1 に格納されたメッセージに従って、プログラムをインストール又は書換え、新たなプログラムに従って制御対象 4 0 0 を制御する。

【 0 0 4 5 】

このようにして、プログラミング装置 1 0 0 にて生成されたメッセージをデータ形式変換装置 2 0 0 で受信し、受信メッセージにて指定された変換タイプに応じた変換処理を実行し、コントローラ 3 0 0 が変換処理をされたメッセージを受信することが可能となる。

【 0 0 4 6 】

したがって、本実施形態のコントローラ管理システムによれば、コントローラ 3 0 0 は、プログラミング装置 1 0 0 が生成したメッセージが XML 形式であるか否かは関知せず、自己が処理可能な言語及び記述等のデータ形式によりメッセージを受信することになる。つまり、コントローラ 3 0 0 が XML 形式に対応可能か否かを問わないことから、本実施形態のコントローラ管理システムによれば既設のコントローラに適用が可能である。

【 0 0 4 7 】

本実施形態では、解釈制御部 2 1 0 とプログラミング装置 1 0 0 とが、また、変換実行部 2 3 0 とコントローラ 3 0 0 とが 1 対 1 で接続されている構成を示した。しかし、図 1 のコントローラ管理システムは、プログラミング装置 1 0 0 とデ

ータ形式変換装置 2 0 0、及びデータ形式変換装置 2 0 0 とコントローラ 3 0 0 をそれぞれ通信ネットワークで結合して構成することができる。この場合、データ形式変換装置 2 0 0 の通信インターフェイスは、プログラミング装置 1 0 0 に対する受信インターフェイスとコントローラ 3 0 0 に対する送信インターフェイスを共通に形成することができる。また、プログラミング装置 1 0 0 とデータ形式変換装置 2 0 0 を一体化し、データ形式変換装置 2 0 0 とコントローラ 3 0 0 を通信ネットワークで結合して構成することもできる。

【 0 0 4 8 】

また、図 1 において、プログラミング装置 1 0 0 とコントローラ 3 0 0 がそれぞれ 1 つの場合を示したが、本発明はこれに限らずプログラミング装置 1 0 0 とコントローラ 3 0 0 が複数の場合にも適用できる。

【 0 0 4 9 】

また、データ形式変換装置 2 0 0 により変換して出力されるメッセージのデータ形式は、記憶装置 2 4 0 に任意の変換プログラムを格納することにより、任意のデータ形式に変換することができる。しかし、テキスト形式に変換して出力する方が、後の管理が容易となる点で好適である。望ましくは、変換後のメッセージも XML 形式とし、かつ簡易な例えば整形形式とすることにより、処理能力の低いコントローラ 3 0 0 の場合であっても、XML 形式の簡単な検査のみで利用することが可能となる。これにより、変換後のメッセージの管理が容易になる。

【 0 0 5 0 】

ところで、メッセージがコントローラ 3 0 0 に格納されるまでに、メッセージの一部が外乱により欠落する場合があり、しかもその欠落を判別できない場合がある。そこで、メッセージの一貫性ないし同一性を検証するために、メッセージの内部にチェックサムを設け、メッセージのテキストデータを単純足し合わせ、あるいは CRC (Cyclic Redundancy Check) 等により演算した結果を、メッセージに合わせてコントローラ 3 0 0 に送信することが好ましい。この場合、コントローラ 3 0 0 において、メッセージ受信後に演算し直して照合することにより、プログラム等の一貫性を検証することができる。特に、チェックサム方式は実装が容易であるから、処理能力の低いコントローラ 3 0 0 の場合にも好適である

【0051】

また、コントローラ300が、異なる複数のデータ形式を受け入れて解釈実行できる場合は、プログラム作成者の意図やデータ形式変換装置200及びコントローラ300の負荷等を考慮して、プログラミング装置100で作成するプログラム等のデータ形式、及びデータ形式変換装置200で変換するコントローラ300用のデータ形式を、それぞれに最適な形式にすることができる。上述の説明では、機械語、XML形式、特定言語形式を例としたが、本発明はそれらのデータ形式に限られないことは自ずと明らかである。

【0052】

要は、コントローラ300が解釈できるデータ形式が機械語に限られる場合は、データ形式変換装置200はプログラミング装置100で作成されたプログラム等を機械語に変換する。コントローラ300がXML形式やHTML形式の解釈能力を備えていれば、データ形式変換装置200はXML形式又はHTML形式に変換する。コントローラ300のデータ形式の解釈能力は、コントローラタイプ情報によって識別するようにすることができる。したがって、データ形式変換装置200は、プログラミング装置100から受信したメッセージに含まれるコントローラタイプ情報に基づいて変換タイプを識別し、識別した変換タイプに対応して設定されている変換データを用いて、プログラミング装置100から受信したメッセージを変換するように構成する。

【0053】

上述したように、本実施形態によれば、データ形式変換装置200は、プログラム等のデータ形式の変換データを備え、プログラミング装置100から出力されるコントローラタイプ情報に基づいてコントローラ300のデータ形式に変換する変換データを用いて、プログラミング装置100から受信したメッセージを変換して、コントローラ300に転送するようにしていることから、次のような効果が得られる。

(1) データ形式変換装置200を開発したことから、プログラミング装置100はコントローラ300のデータ形式に関わらず、例えばプログラム等の作成者

が使い易いマークアップ言語を用いてコントローラのプログラム等を作成することができる。つまり、プログラミング装置 1 0 0 は、計算機と親和性が高く、かつ可読性も高いデータ形式を採用することができ、マークアップ言語に限らず使い易いデータ形式の変更にも容易に対応することができる。

(2) コントローラ 3 0 0 が既設の古いタイプであっても、組み込み計算機の処理能力をアップすることなく、例えば最新のプログラミング装置 1 0 0 によりコントローラ 3 0 0 のプログラム等を管理することができる。つまり、コントローラ 3 0 0 に必要とされる機能は、少なくとも従来のコントローラ 3 0 0 が既に備えているメッセージ受信機能だけで足りるから、安価に本発明を実現することができる。その結果、コントローラ 3 0 0 を管理する管理システムの汎用性ないし柔軟性を高めることができる。

(3) 特に、データ形式変換装置 2 0 0 とコントローラ 3 0 0 とを通信ネットワークを介して通信可能に接続すれば、コントローラ 3 0 0 を遠隔から監視したり、監視したり、制御する遠隔制御システムを実現することができる。

(4) また、プログラミング装置 1 0 0 としては、広く一般に行き渡っているオープン性の高い P C を利用することができるから、故障等を考慮した部品管理等も容易である。

(第 2 の実施の形態)

図 6 に本発明の他の実施形態に係る変換実行部の構成を示す。変換実行部 2 5 0 以外の構成は、図 1 の実施形態と同じである。変換実行部 2 5 0 は、実際に変換処理を行う変換処理部 2 5 1 と、変換するルールが記述されているプロファイル 2 5 2 a ~ 2 5 2 x からなる。変換処理部 2 5 1 は、メッセージを受け、プロファイル 2 5 2 a ~ 2 5 2 x に記述された変換ルールに従ってメッセージを変換する。プロファイルは変換タイプに対応して用意される。プロファイルの格納方式として、例えば変換タイプをファイル名として、ファイルシステムに格納すると管理が容易である。あるいは、変換実行部 2 5 0 内部で持つデータベースから、変換タイプと変換プログラムの対応を得た後にプロファイルを参照することで、変換タイプとプロファイルを柔軟に管理できる。このように、本実施形態によれば、変換処理部 2 5 1 の機能を変更することなく、変換タイプごとにプロファ

イルを用意することにより、適用できる各種のコントローラのタイプを増やすことが可能となる。

【0054】

図7にプロファイルの例を示す。本例ではXML形式で記載されたプロファイルの例を示すが、必要な要件を満たすデータ形式であればXML形式に限定されない。また、行頭の数字とコロンは説明のために付与したものであり、XML形式メッセージとしては必要ない。1行目はXML宣言であり、本メッセージがXML形式のデータであることを表すものである。2行目は文書型宣言であり、本メッセージのデータ構造が"someURI/profile.dtd"で表されるファイルに定義されていることを示す。3行目はプロファイル本体の記述開始タグであり、13行目の終了タグと対をなしている。4行目はキーワードを対応させるルールの開始タグであり、7行目の終了タグと対をなしている。5行目、6行目はキーワード"Start"がキーワード"B"で置き換えられることを意味する。同様に、9行目、10行目ではキーワード"Finish"がキーワード"E"で置き換えられることを意味する。

【0055】

本実施形態では、キーワードを1対1で置き換えられる例を示した。一方で、変換処理機能251がキーワードのセマンティックスを解釈し、外部データベースを参照して対応する文字列に置き換えるという処理を適用することも可能である。セマンティックスを解釈することで、キーワードの幅を吸収するプロファイルとなる。そのためプロファイル数の削減、ひいては、プロファイルを保存する記憶装置の容量削減につながる。

【0056】

図7に示すデータ構造を採用することにより、プログラミング装置100が送信するメッセージに対応して、コントローラタイプ毎に必要な変換データベース（プロファイル）を、容易に準備することが可能となる。なお、図7に示したデータ構造において、本発明で必要な情報が存在する限り、データの追加、削除、変更は可能である。例えば、2行目の文書型宣言は、データが整形式（well formed）であることのみ要求する場合には削除可能である。

【 0 0 5 7 】

上述したように、本実施形態によれば、図 1 の実施形態と同一の効果が得られる。

(第 3 の実施の形態)

図 8 に、本発明の他の実施形態に係るコントローラ管理システムの全体構成を示す。本実施形態が図 1 の実施形態と異なる点は、コントローラ 3 0 0 がサーバ 3 1 0 と記憶装置 3 2 0 を備え、あるいはコントローラ 3 0 0 の外部にサーバ 4 2 0 と記憶装置 4 3 0 が用意され、これらのサーバ 3 1 0、4 2 0 は通信ネットワーク 4 1 0 を介して変換データ取得部 2 2 0 と通信可能に形成されていることにある。なお、本実施形態では、サーバ 3 1 0 とサーバ 4 2 0 の少なくとも一方を備えていればよい。

【 0 0 5 8 】

記憶装置 3 2 0 又は記憶装置 4 3 0 には、前述した実施形態の変換プログラムあるいはプロファイル（以下、変換オブジェクトと呼ぶ。）が格納されている。サーバ 3 1 0 又はサーバ 4 2 0 は、要求に応じて、それぞれ記憶装置 3 2 0 又は記憶装置 4 3 0 から変換オブジェクトを転送する機能を有する。記憶装置 3 2 0 は、コントローラ 3 0 0 に対して必要な変換オブジェクトを少なくとも一つ有する。なお、記憶装置 4 3 0 に複数種類の変換オブジェクトを格納することにより、コントローラ 3 0 0 毎に変換オブジェクトを保持する必要がなくなる。

【 0 0 5 9 】

次に、本実施形態の処理流れを図 9 に示したフローチャートを用いて説明する。本実施形態における処理は、変換データ取得部 2 2 0 の動作を除き、図 5 と同一である。図 9 は、変換データ取得部 2 2 0 の動作を示している。変換データ取得部 2 2 0 は、変換タイプの指定（事象 5 0 2）が入力されるまで待機するブロックである（処理 5 1 0）。変換データ取得部 2 2 0 が変換タイプの指定（事象 5 0 2）を受けて変換実行部を設定する際に、対応する変換オブジェクトが自己の記憶装置 2 4 0 に存在するか否か確認する（処理 5 1 1）。変換オブジェクトが存在しない場合、変換データ取得部 2 2 0 は、ネットワーク 4 1 0 を介してサーバ 3 1 0 又はサーバ 4 2 0 に対して変換オブジェクトを要求する（処理 5 1 2

）。サーバ 3 1 0 又はサーバ 4 2 0 の指定は、プログラミング装置 1 0 0 から送信するメッセージにおいて、例えば図 2 の 6 行目のメッセージの送信先、あるいは、変換タイプのタグの属性として指定されることが望ましい。変換オブジェクトを要求されたサーバ 3 1 0 又はサーバ 4 2 0 は、要求された変換オブジェクトを返信する。ここで、要求された変換オブジェクトが当該サーバに存在しない場合は、当該サーバが他のサーバに要求をリダイレクトするようにすることができる。これにより、サーバ 3 1 0 又は 4 2 0 の管理が柔軟になる。変換データ取得部 2 2 0 が変換オブジェクトを受信した場合、あるいは自己の記憶装置 2 4 0 に変換オブジェクトが存在する場合、変換データ取得部 2 2 0 はその変換オブジェクトを用いて変換実行部 2 3 0 を設定する（事象 5 0 4）。以降は、図 5 の事象 5 0 5 以降と同様である。

【 0 0 6 0 】

本実施形態によれば、データ形式変換装置 2 0 0 に変換タイプに対応する変換オブジェクトが存在しない場合でも、データ形式の変換を行なうことができる。また、コントローラ 3 0 0 がデータ形式の変換オブジェクトを保持することにより、プログラミング装置 1 0 0 に依存しないプログラム方式を実現することができる。また、例えば、コントローラ 3 0 0 のメーカーが、変換オブジェクトを保有するサーバ 4 2 0 を管理するようにすれば、コントローラ 3 0 0 を利用するユーザはコントローラ 3 0 0 内に変換オブジェクトを保持する記憶装置 3 2 0 を持つ必要がない。

【 0 0 6 1 】

さらに、本実施形態によれば、データ形式変換装置 2 0 0 において記憶装置 2 4 0 の実装を省略することができ、データ形式変換装置 2 0 0 の構成を単純化して、安価にすることができる。また、コントローラ 3 0 0 には、変換オブジェクトを送信する機能を追加するだけで、データ形式変換装置 2 0 0 を柔軟に運用することが可能となる。

（第 4 の実施の形態）

図 1 0 に、本発明の更に他の実施の形態に係るデータ形式変換装置 2 0 0 の詳細構成の一例を示す。図に示すように、データ形式変換装置 2 0 0 は、プログラ

ムを格納し実行するメモリ 6 0 0 と、一連の処理を実行するプロセッサ 6 1 0 と、プログラムやオペレーティングシステム（OS）を格納する不揮発記憶装置 6 2 0 と、通信ネットワーク 6 5 0 とインターフェースをとる通信インターフェース（IF） 6 3 0 と、これら要素の間でデータ転送するデータバス 6 4 0 とを含んで形成されている。

【 0 0 6 2 】

メモリ 6 0 0 は、XML パーサ 6 0 1、解釈プログラム 6 0 2、HTTP サーバ 6 0 3、取得プログラム 6 0 4、変換プログラム 6 0 5、OS 6 0 6 を保持するとともに、これらのプログラムで使用するスタックやヒープ、バッファ（図示省略）を保持している。組み込み計算機が利用される分野では、メモリの使用量を減らすために、不揮発記憶装置 6 2 0 にプログラムの命令データを有する場合もある。本実施の形態では、メモリ上にプログラムの命令データを展開する例を示すが、本発明はプログラムがメモリ 6 0 0 あるいは不揮発記憶装置 6 2 0 のいずれにあっても実施可能である。プロセッサ 6 1 0 は、メモリ上のプログラムの命令データを読み込み実行し、所望の機能を実現する。また、プログラミング装置 1 0 0 とコントローラ 3 0 0 と通信するために、プロセッサ 6 1 0 は通信 IF 6 3 0 と通信ネットワーク 6 5 0 を介して通信を行うようになっている。

【 0 0 6 3 】

解釈プログラム 6 0 2 と HTTP サーバ 6 0 3、XML パーサ 6 0 1 は、図 1 又は図 8 の解釈制御部 2 1 0 の機能に対応する。同様に、取得プログラム 6 0 4 は変換データ取得部 2 2 0、変換プログラム 6 0 5 は変換実行部 2 3 0 に相当する。また、不揮発記憶装置 6 2 0 は記憶装置 2 4 0 の機能を兼ねる。OS 6 0 6 は、これらのプログラムを実行するための基本ソフトウェアであり、ハードウェアを制御するドライバや、通信プロトコルのソフトウェア群、プログラムの資源を管理するスケジューラ等から成る。

【 0 0 6 4 】

次に、図 1 0 に示すデータ形式変換装置 2 0 0 の動作を説明する。HTTP サーバ 6 0 3 では、プログラミング装置 1 0 0 からのメッセージを受信し、解釈プログラム 6 0 2 を起動する。解釈プログラム 6 0 2 は、HTTP サーバ 6 0 3 の

CGIプログラムとして実装されるのが好適である。メッセージを受けた解釈プログラム602は、XML形式で記述されたメッセージをXMLパーサ601に通知し、妥当性検証及び解釈を行う。そして、解釈の結果抽出された変換タイプを取得プログラム604に渡す。

【0065】

取得プログラム604は、不揮発記憶装置620を検索して、対応する変換タイプの変換オブジェクトを獲得する。対応する変換タイプの変換オブジェクトが不揮発記憶装置620に存在しなければ、図9のフローチャートに従い、通信ネットワーク650を介して、データ形式変換装置200の外部から取得する。その後、取得プログラム604は取得した変換オブジェクトを用いて、変換プログラム605の設定を行う。取得プログラム604により変換プログラム605の設定完了が解釈プログラム602に通知されると、解釈プログラム602は変換プログラム605に受信したメッセージを通知し、受信されたメッセージの変換を依頼する。変換されたメッセージは、変換プログラム605よりコントローラ300に送信される。

【0066】

変換プログラム605としては、プロセッサ610の機械語にて記述された直接実行可能プログラムのみならず、プラットフォーム独立の中間言語で記述された変換プログラムを利用しても良い。例えば、変換プログラム605を、Java（登録商標）等の仮想機械（VM）と、仮想機械上で実行されるJavaバイトコード等の中間言語プログラムにより構成する。そして、コントローラ300に依存する部分を中間言語プログラムで実装することで、コントローラ300に対応する変換プログラムをデータ形式変換装置200のプロセッサアーキテクチャ毎に用意する必要がなくなる。この場合の中間言語プログラムは、図8に示したように、コントローラ300の記憶装置320又は外部の記憶装置430に格納する。

【0067】

このように構成されることから、本実施形態によれば、図1の実施形態と同一の効果が得られる他、遠隔からコントローラ300を管理することができる。また、利用可能なプログラミング装置として、プラットフォーム独立な言語を実行

可能な計算機とすることで、計算機アーキテクチャやOSに依存することなくプログラミング装置を構成できる。その結果、コントローラ300のユーザは、長期に渡り安定してプログラムの管理をすることができる。

(第5の実施の形態)

図11に、本発明の更に他の実施の形態に係るコントローラ管理システムの全体構成を示す。本実施形態は、前述した他の実施形態に比べて次の点が異なる。すなわち、コントローラ300は記憶装置301とは別に記憶装置330を有する。この記憶装置330は、プログラミング装置100から送信されるデータを格納可能に形成され、変換実行部230'は記憶装置330からデータを取得可能に構成されている。

【0068】

図12に、本実施形態の処理流れのシーケンス図を示す。なお、図12のシーケンス図における事象に付加された番号は、断りのない限り図5に示した事象と同一である。図示のように、プログラミング装置100は、作成したプログラム部分をコントローラ300の記憶装置330に送信する(事象520)。その後、プログラム部分を省略したメッセージをデータ形式変換装置200に送信する(事象500)。ここで、メッセージには、プログラム等が外部に格納される旨の情報を含める。望ましくは、プログラム部分を示すタグの属性として、プログラムが格納されるURL (Uniform Resource Locator) の情報を含める。その後の事象501から事象507は、図5の対応する事象と同様である。受信メッセージの通知を事象507で受けた変換実行部230'は、受信したメッセージによって外部にプログラムが格納されている旨を認識し、コントローラ300にプログラムを要求する(事象521)。その後、解釈制御部210から通知されたメッセージと、取得したプログラムとから変換実行部230'にてプログラムの変換を行い、メッセージをコントローラ300に送信する(事象508)。

【0069】

コントローラ300を制御するプログラム等は、作成した後にも管理のためにプログラム原本が必要となることが多々ある。そのような場合でも本実施形態によれば、作成したプログラム原本をコントローラ自身が保有することになるので

、プログラムを開発したプログラミング装置 1 0 0 だけでなく、他のプログラミング装置においても、コントローラ 3 0 0 にアクセスしてデータ形式を逆変換することなく、プログラムの変更、参照が可能となる。

【 0 0 7 0 】

上述したように、本実施形態によれば、図 1 の実施形態の効果に加えて、プログラミング装置 1 0 0 にて生成されたプログラム等の原本、望ましくは XML 形式にて記述されたプログラム原本を、コントローラ 3 0 0 に保存するようにしているので、プログラミング装置 1 0 0 が異なる装置に代っても、長期に渡ってコントローラ 3 0 0 のプログラム等の管理が容易になる。

（第 6 の実施の形態）

図 1 3 に、本発明の他の実施の形態に係るコントローラ管理システムの全体構成を示す。本実施の形態は、先に説明した実施形態の構成と比べて次の点が異なる。すなわち、プログラミング装置 1 0 0 a ~ 1 0 0 n はブラウザ 1 1 1 を有して構成される。また、プログラミング装置 1 0 0 a ~ 1 0 0 n はデータ形式変換装置の機能を備えて構成される。さらに、コントローラ 3 0 0 は、記憶装置 3 0 1、記憶装置 3 3 0、サーバ 3 1 0 と記憶装置 3 2 0 を備えて構成される。また、複数のプログラミング装置 1 0 0 a ~ 1 0 0 n は、通信ネットワーク 6 5 1 を介してコントローラ 3 0 0 に接続されている。

【 0 0 7 1 】

本実施形態の処理流れを、図 1 4 に示すシーケンス図に従って説明する。図 1 4 における事象に付加された番号は、断りのない限り図 1 2 における事象と同一である。まず、プログラミング装置 1 0 0 a はブラウザを起動する（事象 5 4 0）。その後ユーザの指示（例えば、URL の入力）により、エディタ 1 1 0 のプログラムをコントローラ 3 0 0 へ要求する（事象 5 4 1）。ここで、エディタ 1 1 0 が Java バイトコード等の中間言語プログラムで用意されていれば、コントローラ 3 0 0 にてプログラミング装置のアーキテクチャ毎に変換プログラムを用意しなくても良いから好適である。その後、取得したエディタ 1 1 0 にて、コントローラ 3 0 0 に対するプログラムを記述する（事象 5 4 2）。事象 5 2 0 から始まり事象 5 0 8 に至るまでの動作は、図 1 2 における事象と同様である。

【0072】

本実施形態によれば、プログラミング装置100a~100nは、コントローラ300のプログラムやコントローラ300に関するデータを持たなくても、コントローラ300のプログラムを作成、編集等することができる。さらに、コントローラ300にエディタ等のプログラム編集手段を格納することにより、異なるプログラミング装置間でコントローラ300を制御するプログラムを管理することが可能となる。例えば、JavaのVMを実行する複数の携帯電話から、一つのコントローラ300のプログラムを管理することが可能である。つまり、エディタや変換オブジェクトをJavaバイトコード等の中間言語プログラムで用意することで、携帯電話のような計算機であっても、プログラミング装置として利用することが可能となる。その結果、場所や時間を選ばずに、コントローラ300を管理することが可能となる。

【0073】

さらには、本実施形態に示す構成より、記憶装置240を省略することが可能となる。これにより、プログラミング装置100をより安価に構成することができる。

(第7の実施の形態)

図15に、本発明に係る一実施の形態のプログラミング装置の編集画面に関する一例を示す。メインウィンドウ700は、エディット画面701とコントローラタイプセレクタ702を有する。コントローラのプログラム等を作成するユーザは、エディット画面701上でコントローラ300を制御するプログラム等を編集する。例えば、IEC61131-3に準拠した制御記述言語を利用することができる。又は、テキストで記述する等が考えられる。さらに、ユーザは、エディタが生成する標準形式のプログラムを、どのコントローラに適するタイプに変換するか、例えば図示のように、「PLC-A、PLC-B、PLC-C、… PLC-X」の中から「PLC-B」をコントローラタイプセレクタ702で指定する。コントローラタイプセレクタ702で選んだタイプは、直接的あるいは間接的に図2に示したコントローラタイプを示すデータに反映される。

【0074】

【発明の効果】

以上述べたように、本発明のデータ形式変換装置によれば、プログラム等のデータ形式を異なるデータ形式に変換することができる。

【0075】

また、本発明のコントローラ管理システムは、異なる種類のコントローラを管理することができ、高い汎用性ないし柔軟性を有するものとすることができる。

【0076】

また、本発明のコントローラ管理システムによれば、異なる種類のコントローラを遠隔から管理制御することができる。

【図面の簡単な説明】

【図1】

本発明の第1の実施形態に係るコントローラ管理システムの全体構成を示すブロック図である。

【図2】

本発明に係るプログラミング装置から出力されるXML形式のメッセージの一例を示す図である。

【図3】

図2のXML形式メッセージの変形例を示す図である。

【図4】

変換元のメッセージと変換後のメッセージのデータ形式を対比して説明する図である。

【図5】

図1のコントローラ管理システムの動作を説明するシーケンス図である。

【図6】

本発明の第2の実施形態に係る変換実行部の具体的な構成を示す図である。

【図7】

図7の変換実行部で使用する変換プロファイルの一例を示す図である。

【図8】

本発明の第3の実施形態に係るコントローラ管理システムの全体構成を示すブ

ロック図である。

【図 9】

図 8 の実施形態に係る変換データ取得部の処理手順を示すフローチャートである。

【図 1 0】

本発明の第 4 の実施形態に係るコントロール管理システムの全体構成を示し、特にデータ形式変換装置の詳細な構成を示す図である。

【図 1 1】

本発明の第 5 の実施形態に係るコントロール管理システムの全体構成を示す図である。

【図 1 2】

図 1 1 のコントローラ管理システムの動作を説明するシーケンス図である。

【図 1 3】

本発明の第 6 の実施形態に係るコントロール管理システムの全体構成を示す図である。

【図 1 4】

図 1 3 のコントローラ管理システムの動作を説明するシーケンス図である。

【図 1 5】

本発明の第 7 の実施形態に係るプログラミング装置のプログラム編集画面の一例である。

【符号の説明】

1 0 0、1 0 0 a ～ 1 0 0 n … プログラミング装置

1 1 0 … エディタ

1 1 1 … ブラウザ

2 0 0 … データ形式変換装置

2 1 0 … 解釈制御部

2 2 0 … 変換データ取得部

2 3 0、2 3 0' … 変換実行部

2 4 0 … 記憶装置

250…変換実行部

251…変換処理機能

252a～232x…プロファイル

300…コントローラ

301、320、330、430…記憶装置

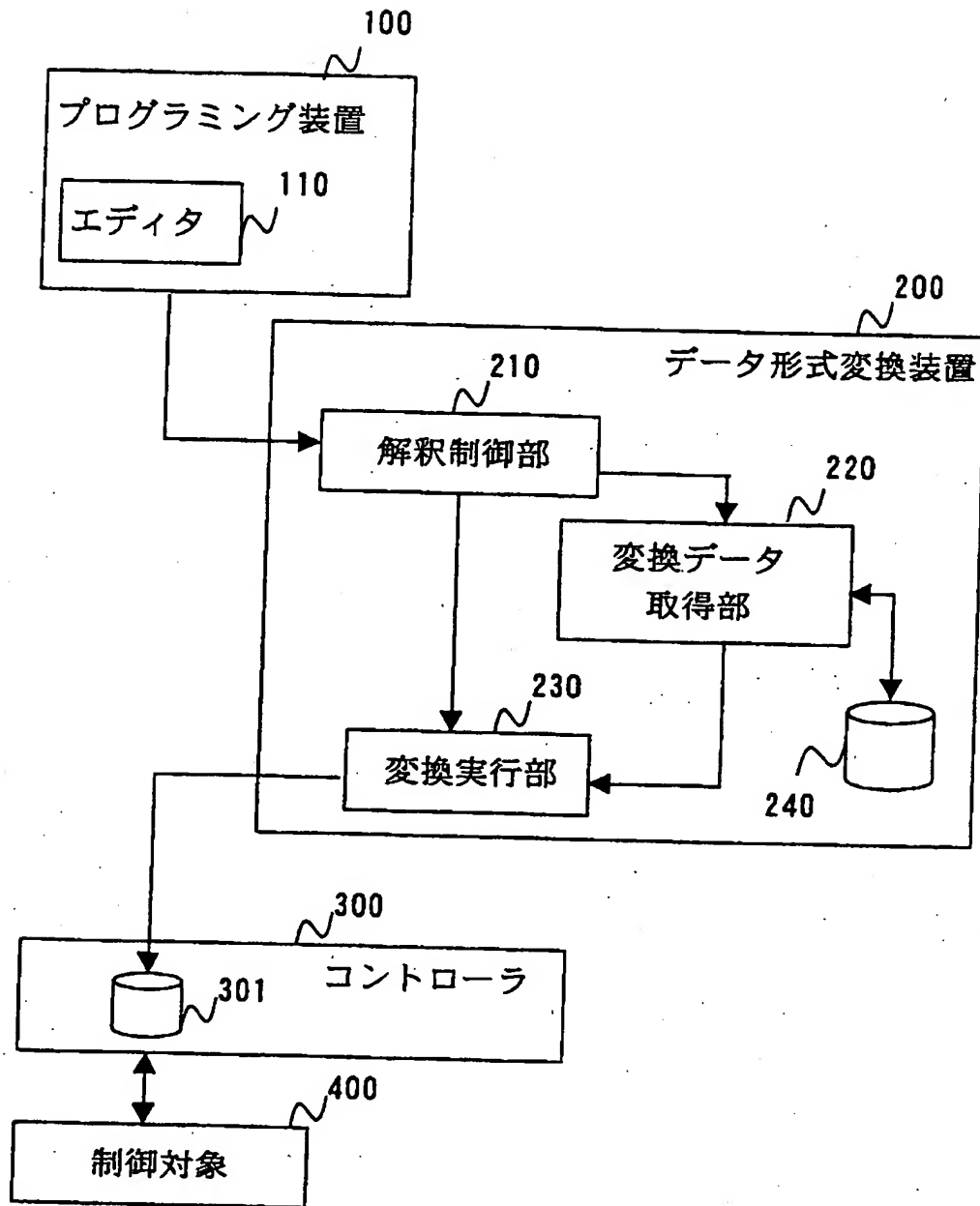
310、420…サーバ

400…制御対象、

410…通信ネットワーク

【書類名】 図面

【図1】



【図 2】

```

1:  <?xml version="1.0"?>
2:  <!DOCTYPE PLCSRC SYSTEM "someURI/plc.dtd">
3:  <PLCSRC>
4:    <DESCRIPTION>
5:      <TYPE>CTL-A</TYPE>
6:      <DEST>192.168.1.1</DEST>
7:      <SRC>192.168.1.2</SRC>
8:    </DESCRIPTION>
9:    <PRG>
10:      <!-- Program HERE -->
11:    </PRG>
12:  </PLCSRC>

```

【図 3】

```

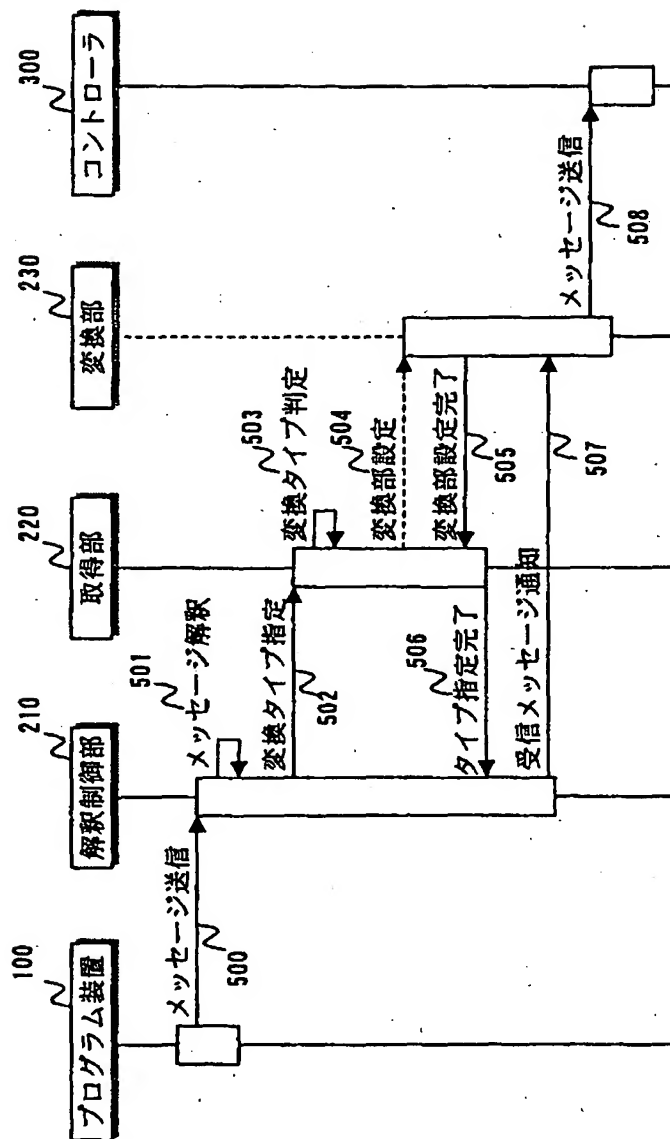
5:  <TYPE SRC="XML-C" DEST="MAC-4">EQ-XY</TYPE>

```

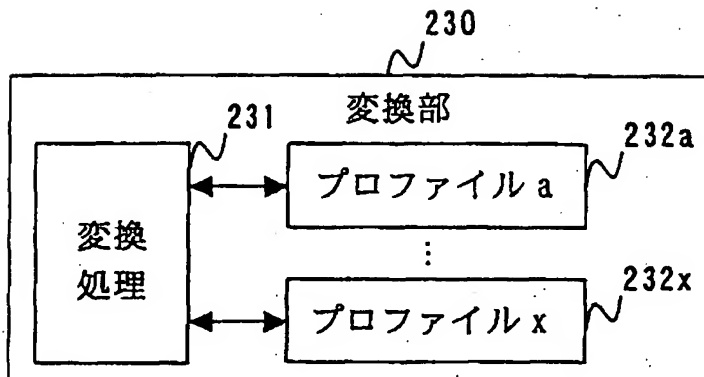
【図 4】

120 ~	121 ~	122 ~	123 ~
タイプ名	対象コントローラ	変換元 形式	変換後 形式
CTL-A	A 型コントローラ	MAC-1	MAC-2
EQ-XY	XY 仕様準拠装置	MAC-3 XML-C LANG-D	MAC-4 XML-E

【図 5】



【図 6】



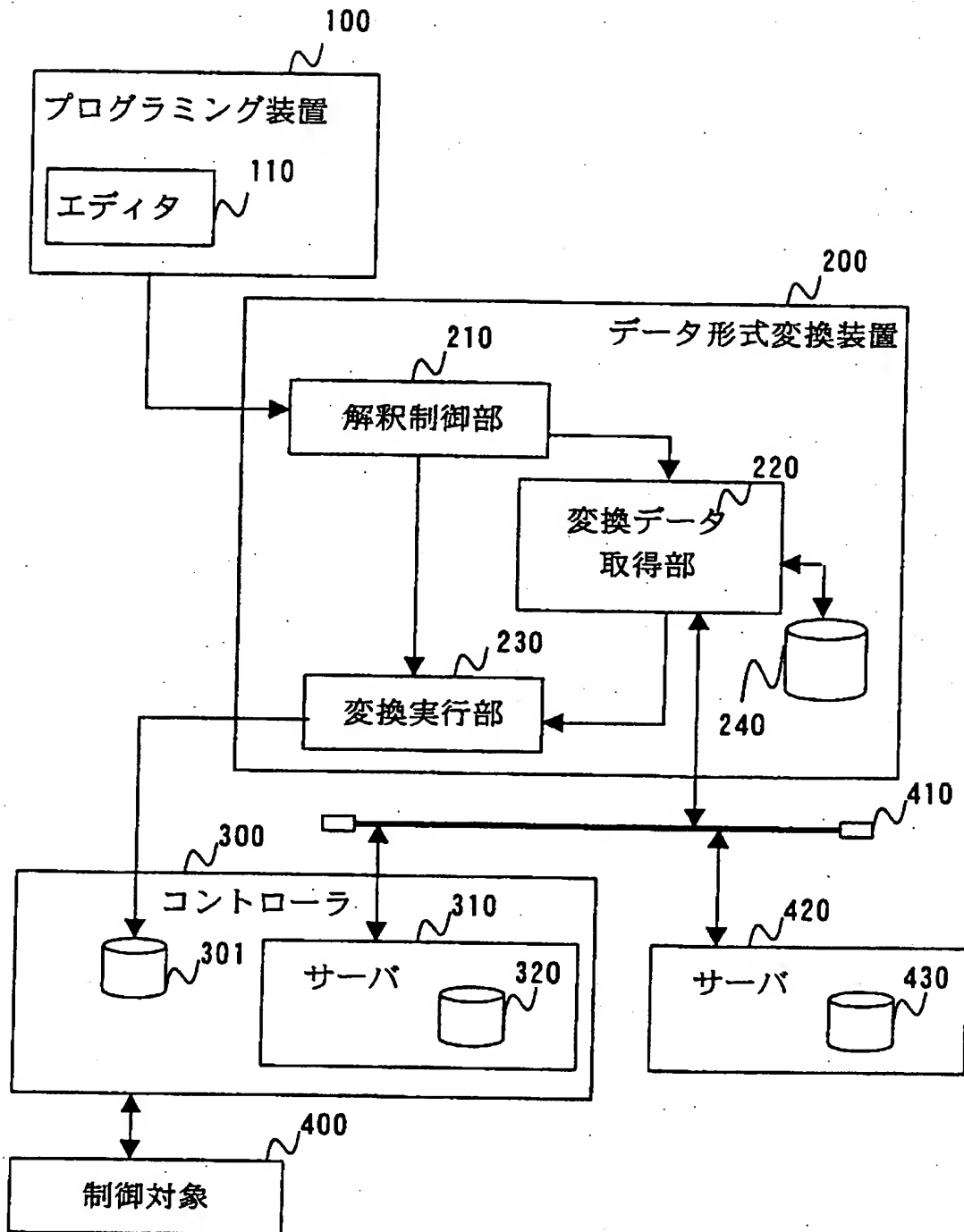
【図 7】

```

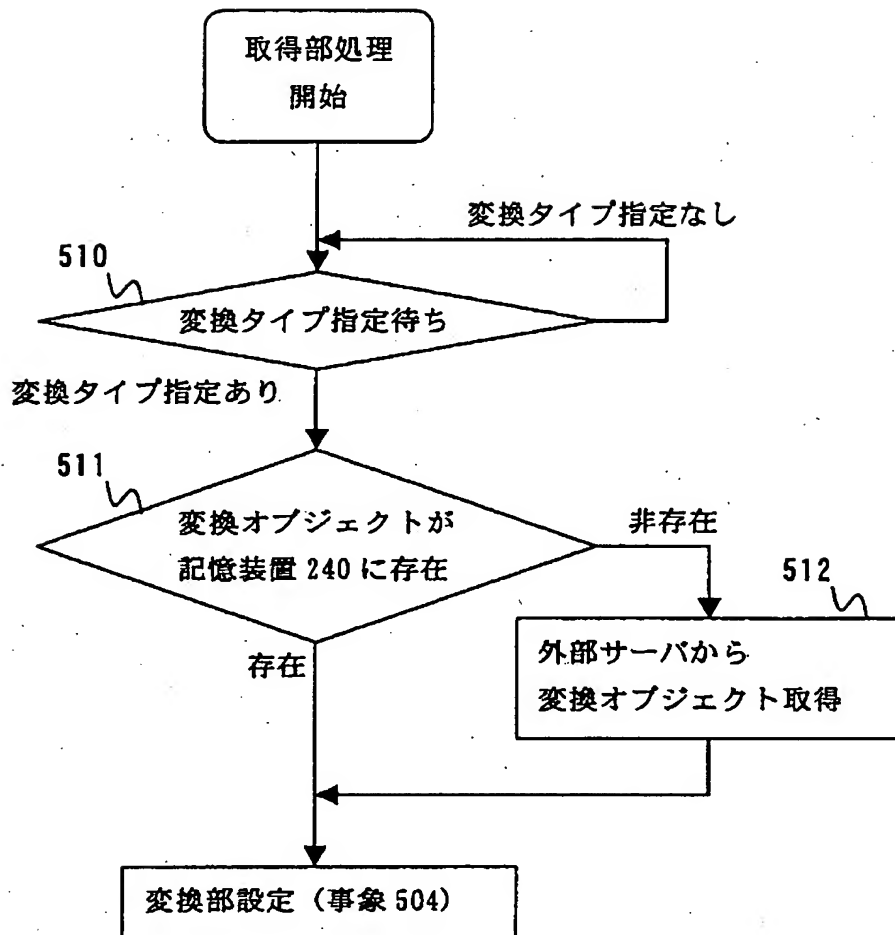
1:  <?xml version="1.0"?>
2:  <!DOCTYPE PROFILE SYSTEM "someURI/profile.dtd">
3:  <PROFILE>
4:    <RULE>
5:      <ORG>Start</ORG>
6:      <CNV>B</CNV>  <!-- B stands for Begin -->
7:    </RULE>
8:    <RULE>
9:      <ORG>Finish</ORG>
10:     <CNV>E</CNV>  <!-- E stands for End -->
11:   </RULE>
12:   ...
13: </PROFILE>

```

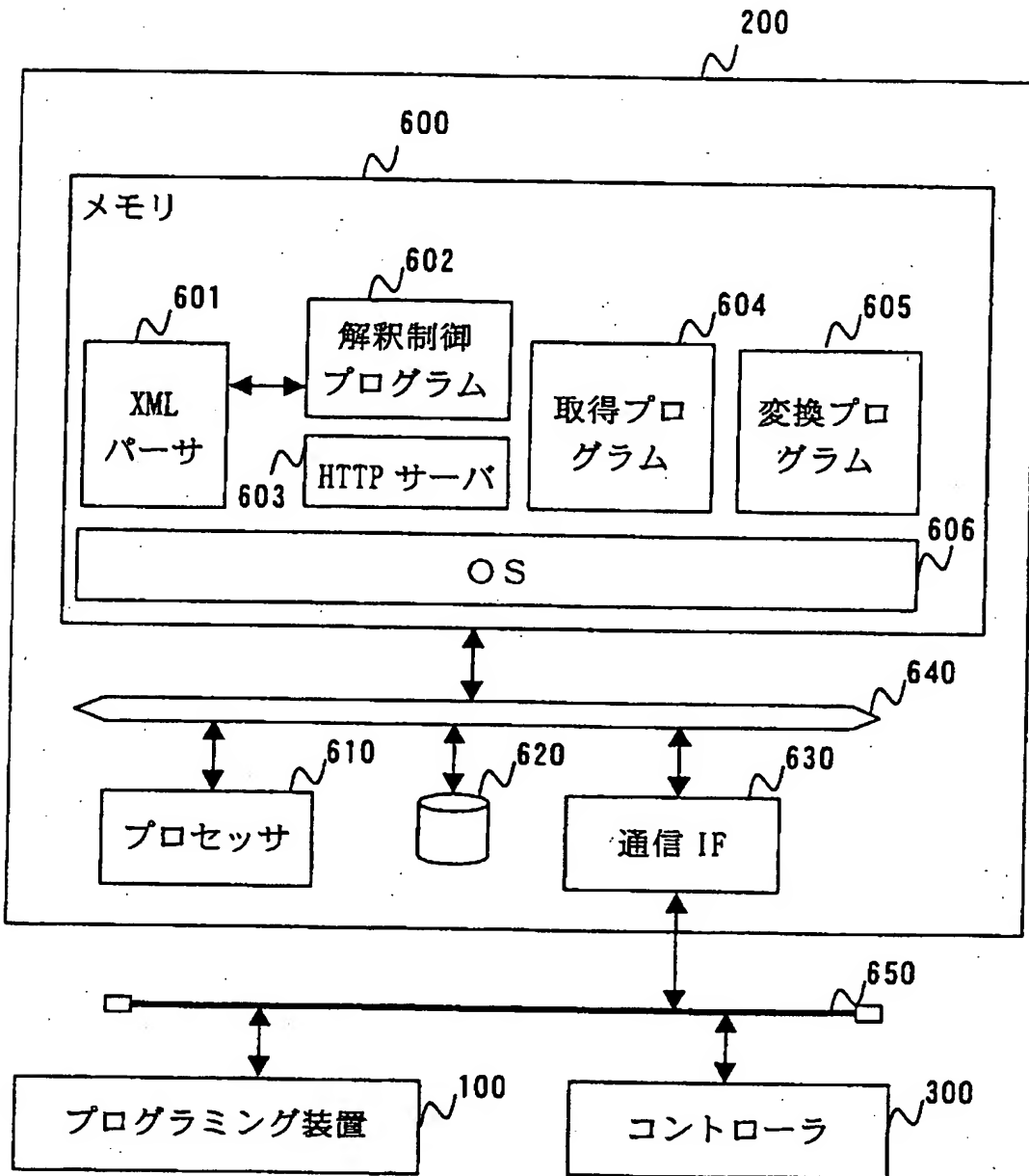
【図 8】



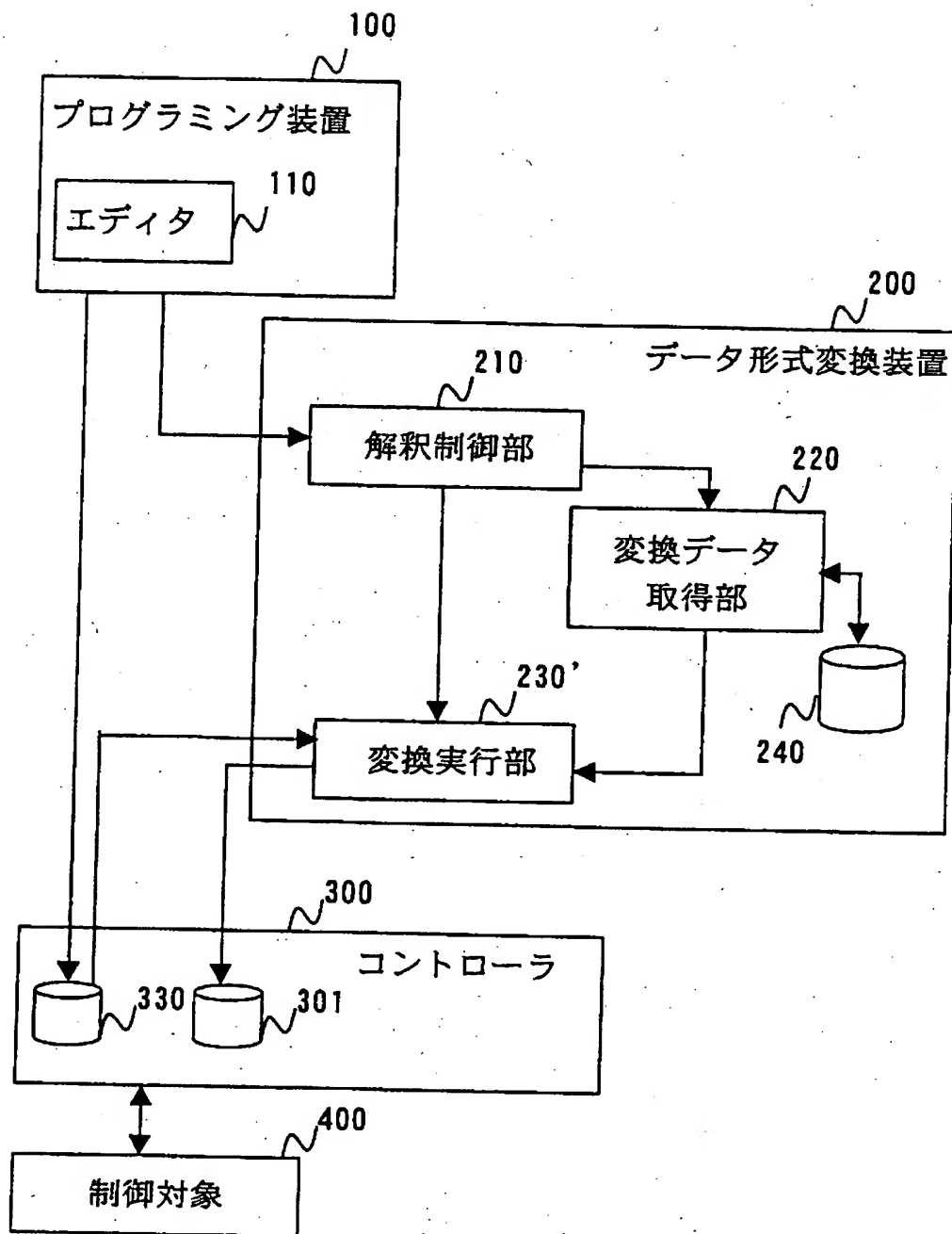
【図 9】



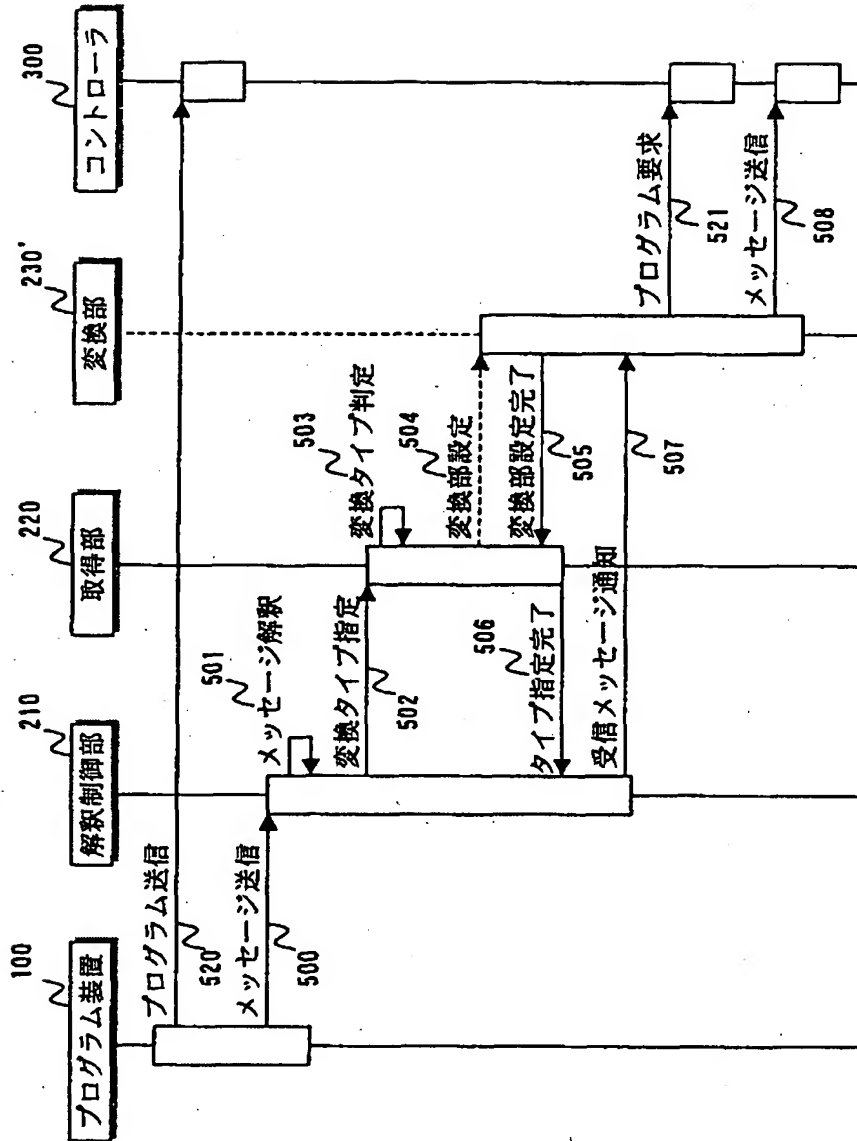
【図10】



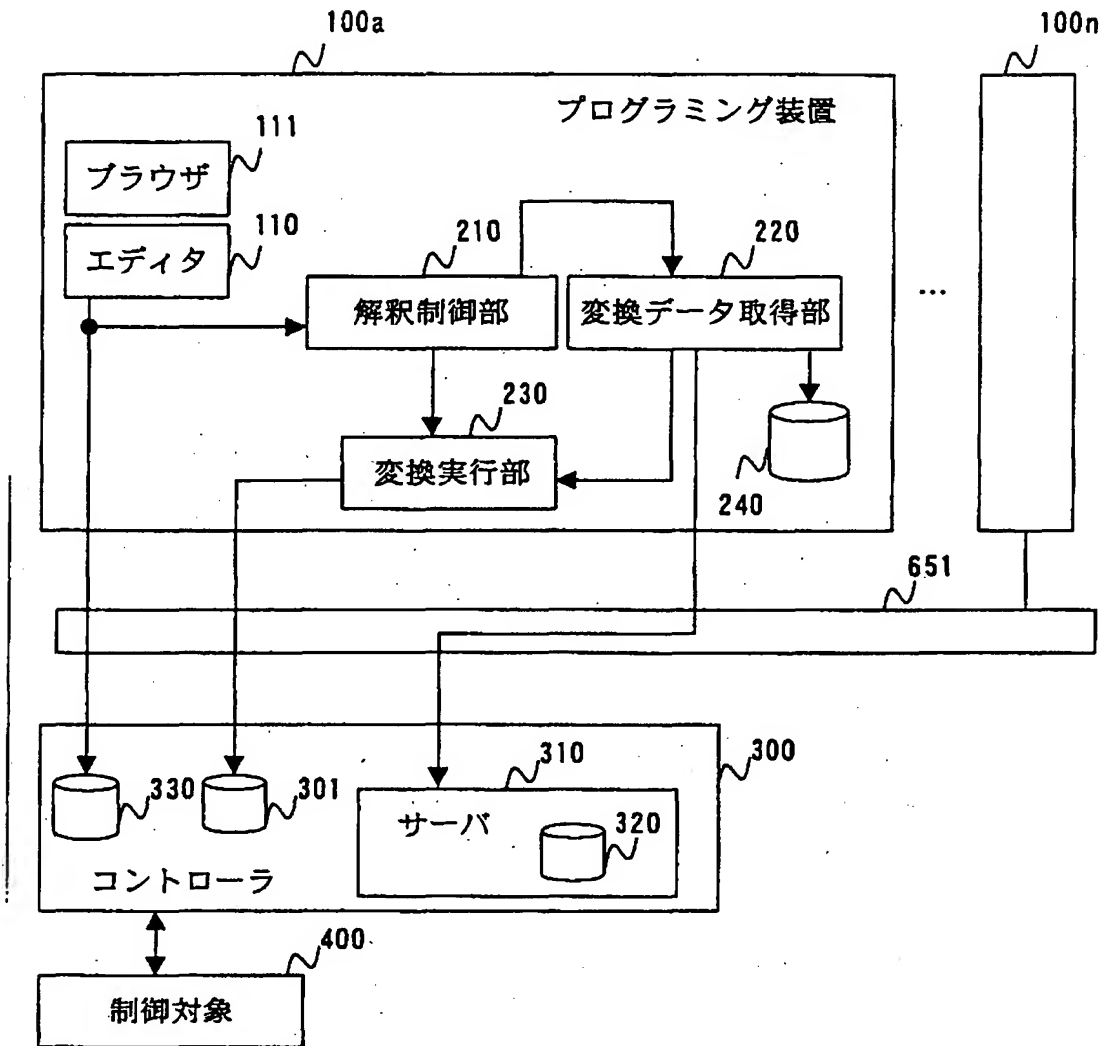
【図 11】



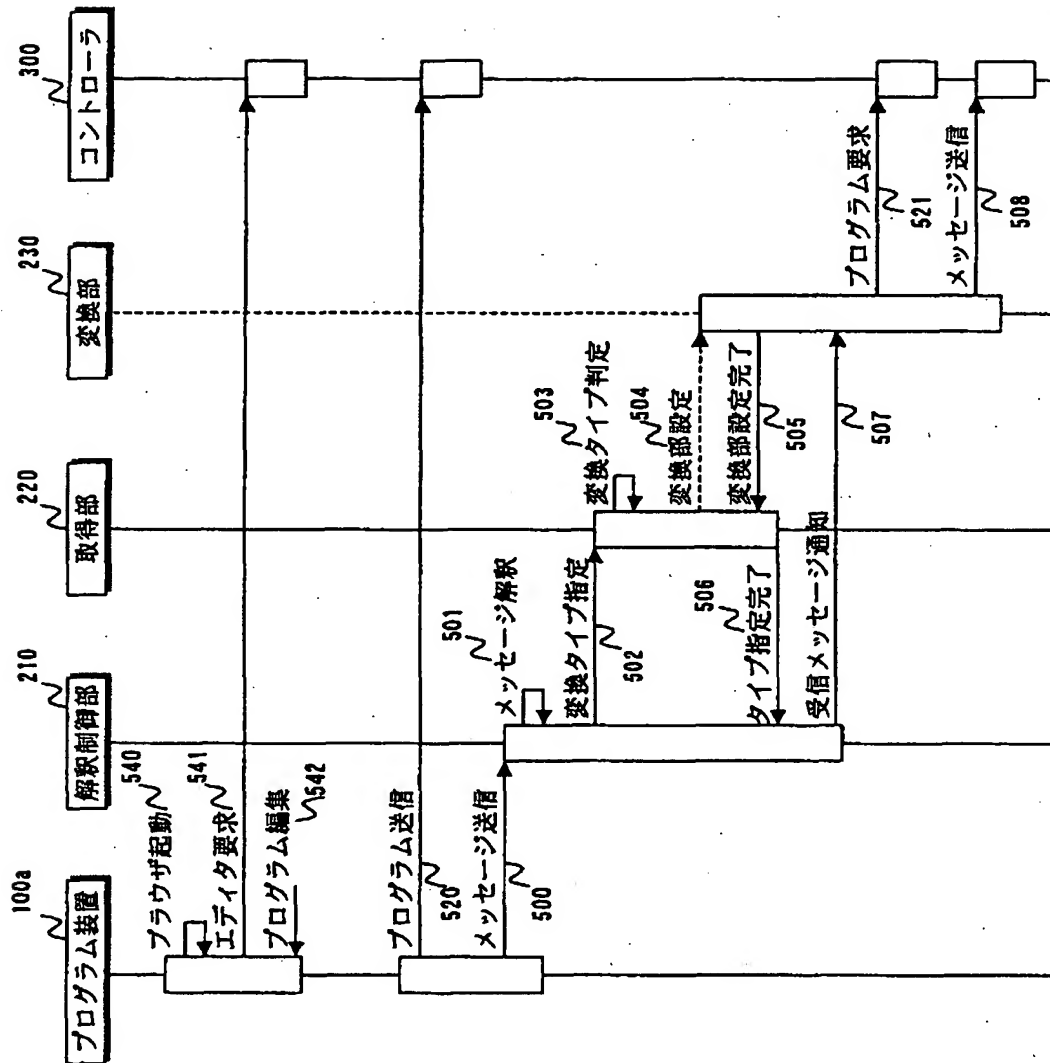
【図 12】



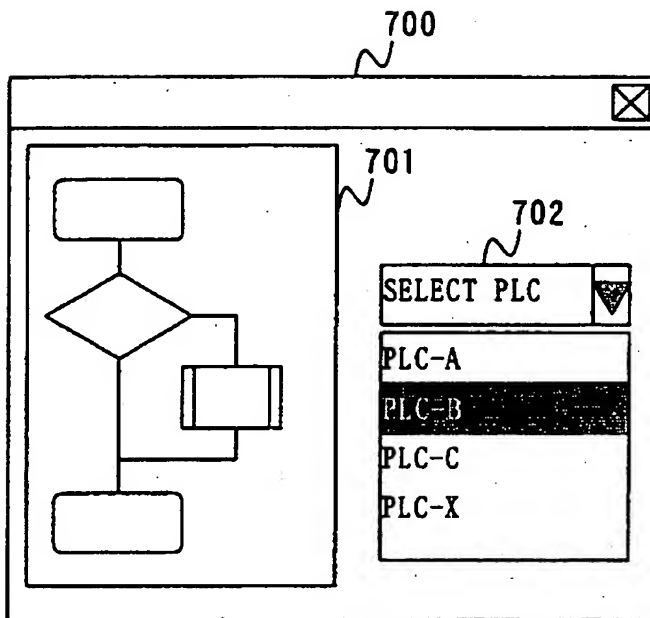
【図 13】



【図14】



【図15】



【書類名】 要約書

【要約】

【課題】 コントローラを管理するシステムの汎用性ないし柔軟性を高める。

【解決手段】 プログラミング装置 1 0 0 から出力される第 1 データ形式のプログラム等を含む入力情報を取り込む受信手段と、取り込まれた入力情報に含まれるデータ形式の変換指令を抽出する解釈手段と、該抽出された変換指令に対応するデータ形式の変換規則を取得する変換データ取得手段と、取得された変換規則に基づいて第 1 データ形式のプログラム等を第 2 データ形式に変換する変換実行手段とを計算機により構成したデータ形式変換装置 2 0 0 を設け、これにより変換した第 2 データ形式のプログラム等をコントローラに出力することにより、コントローラのデータ形式と異なる任意のデータ形式でプログラム等を作成することができるようにしたことを特徴とする。

【選択図】 図 1

出 願 人 履 歴 情 報

識別番号 [000005108]

1. 変更年月日	1990年 8月31日
[変更理由]	新規登録
住 所	東京都千代田区神田駿河台4丁目6番地
氏 名	株式会社日立製作所